

Manipulación de Datos

8

ORACLE®

Copyright © 2004, Oracle. Todos los derechos reservados.

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Describir cada una de las sentencias DML
(Lenguaje de Manipulación de Datos)**
- **Insertar filas en una tabla**
- **Actualizar las filas de una tabla**
- **Suprimir filas de una tabla**
- **Controlar transacciones**

Lenguaje de Manipulación de Datos

- Una sentencia DML se ejecuta al:
 - Agregar nuevas filas a una tabla
 - Modificar filas existentes de una tabla
 - Eliminar filas existentes de una tabla
- Una *transacción* consiste en una recopilación de sentencias DML que forman una unidad lógica de trabajo.

ORACLE

Adición de una Nueva Fila a una Tabla

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

Nueva
fila

Inserción de una fila
nueva en la
tabla DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

ORACLE

Sintaxis de la Sentencia INSERT

- Agregue nuevas filas a una tabla mediante la sentencia INSERT:

```
INSERT INTO table [(column [, column...])]  
VALUES          (value [, value...]);
```

- Con esta sintaxis, sólo se inserta una fila por vez.

ORACLE

Copyright © 2004, Oracle. Todos los derechos reservados.

Inserción de Nuevas Filas

- Inserte una nueva fila que contenga valores para cada columna.
- Enumere valores en el orden por defecto de las columnas de la tabla.
- Opcionalmente, enumere las columnas de la cláusula INSERT.

```
INSERT INTO departments (department_id,  
                      department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);  
1 row created.
```

- Escriba los valores de carácter y de fecha entre comillas simples.



Inserción de Filas con Valores Nulos

- **Método Implícito:** Omita la columna de la lista de columnas.

```
INSERT INTO departments (department_id,  
                        department_name )  
VALUES      (30, 'Purchasing');  
1 row created.
```

- **Método explícito:** Especifique la palabra clave NULL en la cláusula VALUES.

```
INSERT INTO departments  
VALUES      (100, 'Finance', NULL, NULL);  
1 row created.
```

ORACLE

Inserción de Valores Especiales

La función SYSDATE registra la fecha y la hora actuales.

```
INSERT INTO employees (employee_id,
                      first_name, last_name,
                      email, phone_number,
                      hire_date, job_id, salary,
                      commission_pct, manager_id,
                      department_id)
VALUES
(113,
  'Louis', 'Popp',
  'LPOPP', '515.124.4567',
  SYSDATE, 'AC_ACCOUNT', 6900,
  NULL, 205, 100);
```

1 row created.

ORACLE

Copyright © 2004, Oracle. Todos los derechos reservados.

Inserción de Valores de Fecha Específicos

- Agregue un nuevo empleado.

```
INSERT INTO employees  
VALUES      (114,  
              'Den', 'Raphealy',  
              'DRAPHEAL', '515.127.4561',  
              TO_DATE('FEB 3, 1999', 'MON DD, YYYY'),  
              'AC_ACCOUNT', 11000, NULL, 100, 30);  
1 row created.
```

- Verifique la adición.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_P
114	Den	Raphealy	DRAPHEAL	515.127.4561	03-FEB-99	AC_ACCOUNT	11000	

Creación de un Archivo de Comandos

- Utilice la sustitución & en una sentencia SQL para solicitar valores.
- & es una variable pendiente para el valor de variable.

```
INSERT INTO departments
          (department_id, department_name, location_id)
VALUES      (&department_id, '&department_name', &location);
```

Define Substitution Variables

"department_id" 40

"department_name" Human Resources

"location" 2500

Submit for Execution

Cancel

1 row created.

ORACLE

Copia de Filas de Otra Tabla

- Escriba la sentencia INSERT con una subconsulta.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
SELECT employee_id, last_name, salary, commission_pct
FROM employees
WHERE job_id LIKE '%REP%';
```

4 rows created.

- No utilice la cláusula VALUES.
- Haga corresponder el número de columnas de la cláusula INSERT con los de la subconsulta.

Cambio de Datos en una Tabla

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000		60
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000		60
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200		60
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800		50

Actualice las filas de la tabla EMPLOYEES:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000		30
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000		30
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200		30
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800		50

ORACLE

Sintaxis de la Sentencia UPDATE

- **Modifique las filas existentes con la sentencia UPDATE:**

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

- **Actualice más de una fila a la vez (si es necesario).**

Actualización de las Filas de una Tabla

- Se modifican una o varias filas específicas si se especifica la cláusula WHERE:

```
UPDATE employees
SET department_id = 70
WHERE employee_id = 113;
1 row updated.
```

- Se modifican todas las filas de la tabla si se omite la cláusula WHERE:

```
UPDATE copy_emp
SET department_id = 110;
22 rows updated.
```

ORACLE

Actualización de Dos Columnas con una Subconsulta

Actualice el trabajo y el salario del empleado 114 para hacerlos corresponder con los del empleado 205.

```
UPDATE      employees
SET          job_id   =  (SELECT    job_id
                         FROM      employees
                         WHERE     employee_id = 205),
            salary   =  (SELECT    salary
                         FROM      employees
                         WHERE     employee_id = 205)
WHERE        employee_id    =  114;
1 row updated.
```

ORACLE

Actualización de Filas Basándose en Otra Tabla

Utilice subconsultas en sentencias UPDATE para actualizar las filas de una tabla basándose en los valores de otra tabla:

```
UPDATE copy_emp
SET department_id = (SELECT department_id
                      FROM employees
                      WHERE employee_id = 100)
WHERE job_id          = (SELECT job_id
                         FROM employees
                         WHERE employee_id = 200);
1 row updated.
```

Eliminación de una Fila de una Tabla

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

Suprima una fila de la tabla DEPARTMENTS:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

ORACLE®

Copyright © 2004, Oracle. Todos los derechos reservados.

Sentencia DELETE

Puede eliminar filas existentes de una tabla mediante la sentencia DELETE:

```
DELETE [FROM] table  
[WHERE condition];
```

ORACLE

Copyright © 2004, Oracle. Todos los derechos reservados.

Supresión de Filas de una Tabla

- Al especificar la cláusula WHERE, se suprimen filas específicas:

```
DELETE FROM departments  
WHERE department_name = 'Finance';  
1 row deleted.
```

- Al omitir la cláusula WHERE, se suprimen todas las filas de la tabla:

```
DELETE FROM copy_emp;  
22 rows deleted.
```

Supresión de Filas Basándose en Otra Tabla

Utilice subconsultas en sentencias DELETE para eliminar filas de una tabla basándose en valores de otra tabla:

```
DELETE FROM employees
WHERE department_id =
    (SELECT department_id
     FROM departments
     WHERE department_name
           LIKE '%Public%');

1 row deleted.
```

ORACLE

Copyright © 2004, Oracle. Todos los derechos reservados.

Sentencia TRUNCATE

- Elimina todas las filas de una tabla (deja la tabla vacía con la estructura de tabla intacta)
- Es una sentencia DDL (Lenguaje de Definición de Datos) y no DML; no resulta sencillo deshacerla
- Sintaxis:

```
TRUNCATE TABLE table_name;
```

- Ejemplo:

```
TRUNCATE TABLE copy_emp;
```

ORACLE

Uso de Subconsultas en una Sentencia INSERT

```
INSERT INTO
    (SELECT employee_id, last_name,
            email, hire_date, job_id, salary,
            department_id
     FROM   employees
     WHERE  department_id = 50)
VALUES (99999, 'Taylor', 'DTAYLOR',
        TO_DATE('07-JUN-99', 'DD-MON-RR'),
        'ST_CLERK', 5000, 50);
```

1 row created.

ORACLE

Uso de Subconsultas en una Sentencia INSERT

Verifique los resultados:

```
SELECT employee_id, last_name, email, hire_date,  
       job_id, salary, department_id  
FROM   employees  
WHERE  department_id = 50;
```

EMPLOYEE_ID	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
124	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50
141	Rajs	TRAJS	17-OCT-95	ST_CLERK	3500	50
142	Davies	CDAVIES	29-JAN-97	ST_CLERK	3100	50
143	Matos	RMATOS	15-MAR-98	ST_CLERK	2600	50
144	Vargas	PVARGAS	09-JUL-98	ST_CLERK	2500	50
99999	Taylor	DTAYLOR	07-JUN-99	ST_CLERK	5000	50

6 rows selected.

ORACLE®

Copyright © 2004, Oracle. Todos los derechos reservados.

Transacciones de la Base de Datos

Una transacción de base de datos consta de:

- **Sentencias DML que constituyen un cambio consistente en los datos**
- **Una sentencia DDL**
- **Una sentencia DCL (Lenguaje de Control de Datos)**

ORACLE

Transacciones de la Base de Datos

- **Comienzan cuando se ejecuta la primera sentencia DML de SQL**
- **Finalizan con uno de los siguientes eventos:**
 - Se emite una sentencia COMMIT o ROLLBACK.
 - Se ejecuta una sentencia DDL o DCL (validación automática).
 - El usuario sale de *iSQL*Plus*.
 - Se produce un error en el sistema.

ORACLE

Ventajas de las Sentencias COMMIT y ROLLBACK

Con las sentencias COMMIT y ROLLBACK, puede:

- **Garantizar la consistencia de datos**
- **Realizar una presentación preliminar de los cambios de datos antes de hacer que éstos sean permanentes**
- **Agrupar operaciones relacionadas lógicamente**

Control de Transacciones

Tiempo *COMMIT*

Transacción

DELETE

SAVEPOINT A

INSERT

UPDATE

SAVEPOINT B

INSERT

ROLLBACK
to SAVEPOINT B ROLLBACK
to SAVEPOINT A ROLLBACK

Rollback de Cambios hasta un Marcador

- Cree un marcador en una transacción actual utilizando la sentencia SAVEPOINT.
- Haga rollback hasta ese marcador mediante la sentencia ROLLBACK TO SAVEPOINT.

```
UPDATE...
SAVEPOINT update_done;
Savepoint created.
INSERT...
ROLLBACK TO update_done;
Rollback complete.
```

ORACLE

Procesamiento de Transacciones Implícito

- **Se produce una validación automática en las siguientes circunstancias:**
 - Se emite una sentencia DDL
 - Se emite una sentencia DCL
 - Salida normal de *iSQL*Plus*, sin emitir explícitamente sentencias COMMIT o ROLLBACK
- **Se produce un rollback automático tras una terminación anormal de *iSQL*Plus* o un fallo del sistema.**

Estado de los Datos antes de COMMIT o ROLLBACK

- Se puede recuperar el estado anterior de los datos.
- El usuario actual puede revisar los resultados de las operaciones DML mediante la sentencia SELECT.
- Otros usuarios *no pueden* ver los resultados de las sentencias DML del usuario actual.
- Las filas afectadas están *bloqueadas*; los demás usuarios no pueden cambiar los datos en las filas afectadas.

ORACLE

Estado de los Datos después de COMMIT

- **Los cambios de datos se hacen permanentes en la base de datos.**
- **El estado anterior de los datos se pierde permanentemente.**
- **Todos los usuarios pueden ver los resultados.**
- **Se liberan los bloqueos en las filas afectadas; dichas filas están libres para que las manipulen otros usuarios.**
- **Se borran todos los puntos de grabación.**

ORACLE®

Validación de Datos

- Realice los cambios:

```
DELETE FROM employees  
WHERE employee_id = 99999;  
1 row deleted.
```

```
INSERT INTO departments  
VALUES (290, 'Corporate Tax', NULL, 1700);  
1 row created.
```

- Valide los cambios:

```
COMMIT;  
Commit complete.
```

ORACLE

Estado de los Datos después de ROLLBACK

Deseche todos los cambios pendientes utilizando la sentencia ROLLBACK:

- Se deshacen los cambios de datos.
- Se restaura el estado anterior de los datos.
- Se liberan los bloqueos en las filas afectadas.

```
DELETE FROM copy_emp;  
22 rows deleted.  
ROLLBACK;  
Rollback complete.
```

ORACLE

Estado de los Datos después de ROLLBACK

```
DELETE FROM test;  
25,000 rows deleted.
```

```
ROLLBACK;  
Rollback complete.
```

```
DELETE FROM test WHERE id = 100;  
1 row deleted.
```

```
SELECT * FROM test WHERE id = 100;  
No rows selected.
```

```
COMMIT;  
Commit complete.
```

ORACLE

Rollback de Nivel de Sentencia

- **Si falla una única sentencia DML durante la ejecución, sólo se hace rollback de esa sentencia.**
- **Oracle Server implementa un punto de grabación implícito.**
- **Se retienen todos los demás cambios.**
- **El usuario debe terminar las transacciones explícitamente ejecutando una sentencia COMMIT o ROLLBACK.**

ORACLE

Consistencia de Lectura

- **La consistencia de lectura garantiza una visualización consistente de los datos en todo momento.**
- **Los cambios realizados por un usuario no entran en conflicto con los realizados por otro.**
- **La consistencia de datos garantiza que en los mismos datos:**
 - Los lectores no tienen que esperar a los escritores
 - Los escritores no tienen que esperar a los lectores

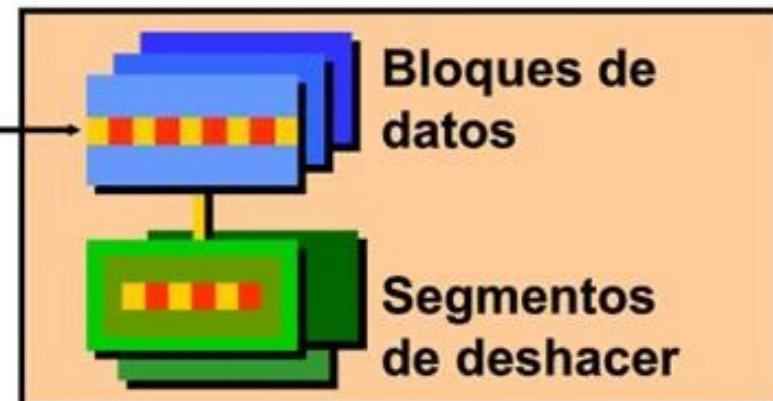
ORACLE

Implementación de la Consistencia de Lectura

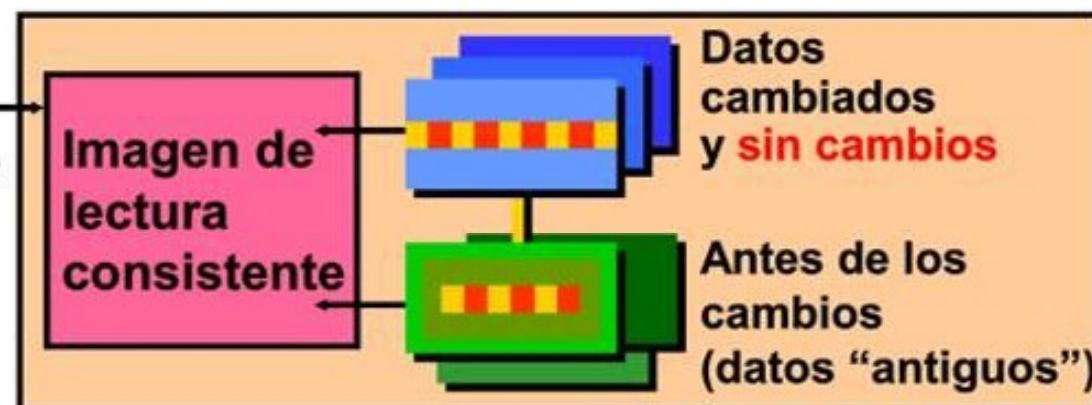
Usuario A



```
UPDATE employees  
SET salary = 7000  
WHERE last_name = 'Goyal';
```



```
SELECT *  
FROM userA.employees;
```



Usuario B

ORACLE

Resumen

En esta lección, ha aprendido a utilizar estas sentencias:

Función	Descripción
INSERT	Agrega una nueva fila a la tabla
UPDATE	Modifica filas existentes de la tabla
DELETE	Elimina filas existentes de la tabla
COMMIT	Hace permanentes todos los cambios pendientes
SAVEPOINT	Se utiliza para hacer rollback al marcador de punto de grabación
ROLLBACK	Desecha todos los cambios de datos pendientes

Práctica 8: Visión General

Esta práctica cubre los temas siguientes:

- Inserción de filas en las tablas**
- Actualización y supresión de las filas de la tabla**
- Control de transacciones**